A Bayesian Approach to Digital Matting

Yung-Yu Chuang[1] Brian Curless[1] David H. Salesin[1,2] Richard Szeliski[2]

[1]Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195
[2]Microsoft Research, Redmond, WA 98052
E-mail: {cyy, curless, salesin}@cs.washington.edu szeliski@microsoft.com
http://grail.cs.washington.edu/projects/digital-matting/

Abstract

This paper proposes a new Bayesian framework for solving the matting problem, i.e. extracting a foreground element from a background image by estimating an opacity for each pixel of the foreground element. Our approach models both the foreground and background color distributions with spatially-varying sets of Gaussians, and assumes a fractional blending of the foreground and background colors to produce the final output. It then uses a maximum-likelihood criterion to estimate the optimal opacity, foreground and background simultaneously. In addition to providing a principled approach to the matting problem, our algorithm effectively handles objects with intricate boundaries, such as hair strands and fur, and provides an improvement over existing techniques for these difficult cases.

1. Introduction

In *digital matting*, a foreground element is extracted from a background image by estimating a color and opacity for the foreground element at each pixel. The opacity value at each pixel is typically called its *alpha*, and the opacity image, taken as a whole, is referred to as the *alpha matte* or *key*. Fractional opacities (between 0 and 1) are important for transparency and motion blurring of the foreground element, as well as for partial coverage of a background pixel around the foreground object's boundary.

Matting is used in order to *composite* the foreground element into a new scene. Matting and compositing were originally developed for film and video production [4], where they have proven invaluable. Nevertheless, "pulling a matte" is still somewhat of a black art, especially for certain notoriously difficult cases such as thin whisps of fur or hair. The problem is difficult because it is inherently underconstrained: for a foreground element over a single background image there are in general an infinite number of interpretations for the foreground's color versus opacity.

In practice, it is still possible to pull a satisfactory matte in many cases. One common approach is to use a background image of known color (typically blue or green) and make certain assumptions about the colors in the foreground (such as the relative proportions of red, green, and blue at each pixel); these assumptions can then be tuned by a human operator. Other approaches attempt to pull mattes from natural (arbitrary) backgrounds, using statistics of known regions of foreground or background in order to estimate the foreground and background colors along the boundary. Once these colors are known, the opacity value is uniquely determined.

In this paper, we survey the most successful previous approaches to digital matting—all of them fairly *ad hoc*—and demonstrate cases in which each of them fails. We then introduce a new, more principled approach to matting, based on a Bayesian framework. While no algorithm can give perfect results in all cases (given that the problem is inherently underconstrained), our Bayesian approach appears to give improved results in each of these cases.

2. Background

As already mentioned, matting and compositing were originally developed for film and video production. In 1984, Porter and Duff [8] introduced the digital analog of the matte—the *alpha channel*—and showed how synthetic images with alpha could be useful in creating complex digital images. The most common compositing operation is the *over* operation, which is summarized by the *compositing equation*:

$$C = \alpha F + (1 - \alpha)B\,, \tag{1}$$

where C, F, and B are the pixel's composite, foreground, and background colors, respectively, and α is the pixel's opacity component used to linearly blend between foreground and background.

The matting process starts from a photograph or set of photographs (essentially composite images) and attempts to extract the foreground and alpha images. Matting techniques differ primarily in the number of images and in what *a priori* assumptions they make about the foreground, background, and alpha.

Blue screen matting was among the first techniques used for live action matting. The principle is to photograph the subject against a constant-colored background, and extract foreground and alpha treating each frame in isolation. This single image approach is underconstrained since, at each pixel, we have three observations and four unknowns. Vlahos pioneered the notion of adding simple constraints to make the problem tractable; this work is nicely summarized by Smith

0-7695-1272-0/01 $10.00 © 2001 IEEE

and Blinn [11]. For example, under the assumption that $.5 \leq a_2 \leq F_b \leq a_2 F_g$, Vlahos constrained the set of equations with:

$$\alpha = 1 - a_1(C_b - a_2 C_g)\,, \tag{2}$$

where C_b and C_g are the blue and green channels of the input image, respectively, and a_1 and a_2 are user-controlled tuning parameters. Additional constraint equations such as this one, however, while easy to implement, are *ad hoc*, require an expert to tune them, and can fail on fairly simple foregrounds.

More recently, Mishima [5] developed a blue screen matting technique based on representative foreground and background samples (Figure 1(e)). In particular, the algorithm starts with two identical polyhedral (triangular mesh) approximations of a sphere in *rgb* space centered at the average value $\overline{B}$ of the background samples. The vertices of one of the polyhedra (the *background polyhedron*) are then repositioned by moving them along lines radiating from the center until the polyhedron is as small as possible while still containing all the background samples. The vertices of the other polyhedron (the *foreground polyhedron*) are similarly adjusted to give the *largest* possible polyhedron that contains *no* foreground pixels from the sample provided. Given a new composite color C, then, Mishima casts a ray from $\overline{B}$ through C and defines the intersections with the background and foreground polyhedra to be B and F, respectively. The fractional position of C along the line segment BF is α.

Under some circumstances, it might be possible to photograph a foreground object against a known but non-constant background. One simple approach for handling such a scene is to take a difference between the photograph and the known background and determine α to be 0 or 1 based on an arbitrary threshold. This approach, known as *difference matting* (see, e.g., [9]) is error prone and leads to "jagged" mattes. Smoothing such mattes by blurring can help with the jaggedness but does not generally compensate for gross errors.

One limitation of blue screen and difference matting is the reliance on a controlled environment or imaging scenario that provides a known, possibly constant-colored background. The more general problem of extracting foreground and alpha from relatively arbitrary photographs or video streams is known as *natural image matting*. To our knowledge, the two most successful natural image matting systems are Knockout, developed by Ultimatte (and, to the best of our knowledge, described in patents by Berman *et al.* [1, 2]), and the technique of Ruzon and Tomasi [10]. In both cases, the process begins by having a user segment the image into three regions: definitely foreground, definitely background, and unknown (as illustrated in Figure 1(a)). The algorithms then estimate F, B, and α for all pixels in the unknown region.

For Knockout, after user segmentation, the next step is to extrapolate the known foreground and background colors into the unknown region. In particular, given a point in the unknown region, the foreground F is calculated as a weighted sum of the pixels on the perimeter of the known foreground region. The weight for the nearest known pixel is set to 1, and this weight tapers linearly with distance, reaching 0 for pixels that are twice as distant as the nearest pixel. The same procedure is used for initially estimating the background B' based on nearby known background pixels. Figure 1(b) shows a set of pixels that contribute to the calculation of F and B' of an unknown pixel.

The estimated background color B' is then refined to give B using one of several methods [2] that are all similar in character. One such method establishes a plane through the estimated background color with normal parallel to the line $B'F$. The pixel color in the unknown region is then projected along the direction of the normal onto the plane, and this projection becomes the refined guess for B. Figure 1(f) illustrates this procedure.

Finally, Knockout estimates α according to the relation

$$\alpha = \frac{f(C) - f(B)}{f(F) - f(B)}\,, \tag{3}$$

where $f(\cdot)$ projects a color onto one of several possible axes through *rgb* space (e.g., onto one of the *r*-, *g*-, or *b*- axes). Figure 1(f) illustrates alphas computed with respect to the *r*- and *g*- axes. In general, α is computed by projection onto all of the chosen axes, and the final α is taken as a weighted sum over all the projections, where the weights are proportional to the denominator in equation (3) for each axis.

Ruzon and Tomasi [10] take a probabilistic view that is somewhat closer to our own. First, they partition the unknown boundary region into sub-regions. For each sub-region, they construct a box that encompasses the sub-region and includes some of the nearby known foreground and background regions (see Figure 1(c)). The encompassed foreground and background pixels are then treated as samples from distributions $P(F)$ and $P(B)$, respectively, in color space. The foreground pixels are split into coherent clusters, and unoriented Gaussians (i.e., Gaussians that are axis-aligned in color space) are fit to each cluster, each with mean $\overline{F}$ and diagonal covariance matrix Σ_F. In the end, the foreground distribution is treated as a mixture (sum) of Gaussians. The same procedure is performed on the background pixels yielding Gaussians, each with mean $\overline{B}$ and covariance Σ_B, and then every foreground cluster is paired with every background cluster. Many of these pairings are rejected based on various "intersection" and "angle" criteria. Figure 1(g) shows a single pairing for a foreground and background distribution.

After building this network of paired Gaussians, Ruzon and Tomasi treat the observed color C as coming from an intermediate distribution $P(C)$, somewhere between the foreground and background distributions. The intermediate distribution is also defined to be a sum of Gaussians, where each Gaussian is centered at a distinct mean value $\overline{C}$ located fractionally (according to a given alpha) along a line between the mean of each foreground and background cluster pair with

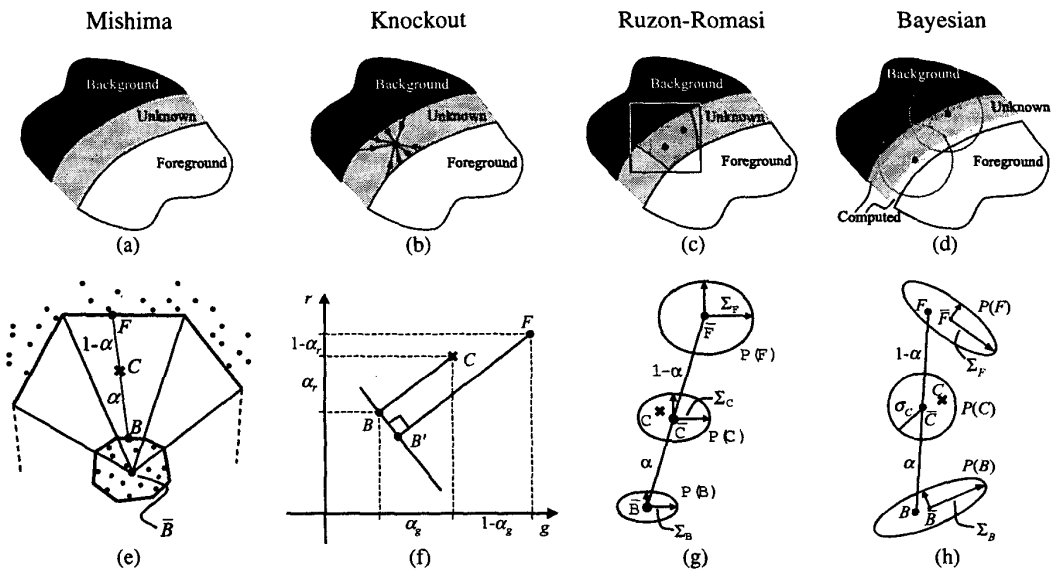

Figure 1 Summary of algorithms. Each of the algorithms shown in this figure requires some specification of background and foreground pixels. Mishima's algorithm (a) uses these samples to form a global distribution, whereas Knockout (b), Ruzon-Tomasi (c), and our new Bayesian approach (d) analyze unknown pixels using local distributions. The dark gray area in (c) corresponds to a segment within the unknown region that will be evaluated using the statistics derived from the square region's overlap with the labeled foreground and background. Figures (e)-(h) show how matte parameters are computed using the Mishima, Knockout, Ruzon-Tomasi, and our Bayesian approach, respectively.

fractionally interpolated covariance Σ_C, as depicted in Figure 1(g). The optimal alpha is the one that yields an intermediate distribution for which the observed color has maximum probability; i.e., the optimal α is chosen independently of F and B. As a post-process, the F and B are computed as weighted sums of the foreground and background cluster means using the individual pairwise distribution probabilities as weights. The F and B colors are then perturbed to force them to be endpoints of a line segment passing through the observed color and satisfying the compositing equation.

Both the Knockout and the Ruzon-Tomasi techniques can be extended to video by hand-segmenting each frame, but more automatic techniques are desirable for video. Mitsunaga *et al.* [6] developed the AutoKey system for extracting foreground and alpha mattes from video, in which a user seeds a frame with foreground and background contours, which then evolve over time. This approach, however, makes strong smoothness assumptions about the foreground and background (in fact, the extracted foreground layer is assumed to be constant near the silhouette) and is designed for use with fairly hard edges in the transition from foreground to background; i.e., it is not well-suited for transparency and hair-like silhouettes.

In each of the cases above, a single observation of a pixel yields an underconstrained system that is solved by building spatial distributions or maintaining temporal coherence. Wallace [12] provided an alternative solution that was independently (and much later) developed and refined by Smith and Blinn [11]: take an image of the same object in front of multiple known backgrounds. This approach leads to an overconstrained system without building any neighborhood distributions and can be solved in a least-squares framework. While this approach requires even more controlled studio conditions than the single solid background used in blue screen matting and is not immediately suitable for live-action capture, it does provide a means of estimating highly accurate foreground and alpha values for real objects. We use this method to provide ground-truth mattes when making comparisons.

3. Our Bayesian framework

For the development that follows, we will assume that our input image has already been segmented into three regions: "background," "foreground," and "unknown," with the background and foreground regions having been delineated conservatively. The goal of our algorithm, then, is to solve for the foreground color F, background color B, and opacity α given the observed color C for each pixel within the unknown region of the image. Since F, B, and C have three color channels each, we have a problem with three equations and seven unknowns.

Like Ruzon and Tomasi [10], we will solve the problem in part by building foreground and background probability

distributions from a given neighborhood. Our method, however, uses a continuously sliding window for neighborhood definitions, marches inward from the foreground and background regions, and utilizes nearby *computed* F, B, and α values (in addition to these values from "known" regions) in constructing oriented Gaussian distributions, as illustrated in Figure 1(d). Further, our approach formulates the problem of computing matte parameters in a well-defined Bayesian framework and solves it using the *maximum a posteriori* (MAP) technique. In this section, we describe our Bayesian framework in detail.

In MAP estimation, we try to find the most likely estimates for F, B, and α, given the observation C. We can express this as a maximization over a probability distribution P and then use Bayes's rule to express the result as the maximization over a sum of log likelihoods:

$$\begin{aligned} &\arg\max_{F,B,\alpha} P(F,B,\alpha \mid C) \qquad (4)\\ &= \arg\max_{F,B,\alpha} P(C \mid F,B,\alpha)\, P(F)\, P(B)\, P(\alpha) \,/\, P(C)\\ &= \arg\max_{F,B,\alpha} L(C \mid F,B,\alpha) + L(F) + L(B) + L(\alpha)\,, \end{aligned}$$

where $L(\cdot)$ is the *log likelihood* $L(\cdot) = \log P(\cdot)$, and we drop the $P(C)$ term because it is a constant with respect to the optimization parameters. (Figure 1(h) illustrates the distributions over which we solve for the optimal F, B, and α parameters.)

The problem is now reduced to defining the log likelihoods $L(C \mid F,B,\alpha)$, $L(F)$, $L(B)$, and $L(\alpha)$.

We can model the first term by measuring the difference between the observed color and the color that would be predicted by the estimated F, B, and α:

$$L(C \mid F,B,\alpha) \;=\; -\|C - \alpha F - (1-\alpha)B\|^2 / \sigma_C^2\,. \quad (5)$$

This log-likelihood models error in the measurement of C and corresponds to a Gaussian probability distribution centered at $\overline{C} = \alpha F + (1-\alpha)B$ with standard deviation σ_C.

We use the spatial coherence of the image to estimate the foreground term $L(F)$. That is, we build the color probability distribution using the known and previously estimated foreground colors within each pixel's neighborhood N. To more robustly model the foreground color distribution, we weight the contribution of each nearby pixel i in N according to two separate factors. First, we weight the pixel's contribution by α_i^2, which gives colors of more opaque pixels higher confidence. Second, we use a spatial Gaussian falloff g_i with $\sigma = 8$ to stress the contribution of nearby pixels over those that are further away. The combined weight is then $w_i = \alpha_i^2 g_i$.

Given a set of foreground colors and their corresponding weights, we first partition colors into several clusters using the method of Orchard and Bouman [7]. For each cluster, we calculate the weighted mean color $\overline{F}$ and the weighted covariance matrix Σ_F:

$$\overline{F} \;=\; \frac{1}{W}\sum_{i\in N} w_i\, F_i \qquad (6)$$

$$\Sigma_F \;=\; \frac{1}{W}\sum_{i\in N} w_i\,(F_i - \overline{F})\,(F_i - \overline{F})^T \qquad (7)$$

where $W = \sum_{i\in N} w_i$. The log likelihoods for the foreground $L(F)$ can then be modeled as being derived from an oriented elliptical Gaussian distribution, using the weighted covariance matrix as follows:

$$L(F) \;=\; -(F - \overline{F})^T\, \Sigma_F^{-1}\, (F - \overline{F})\,/\,2\,. \qquad (8)$$

The definition of the log likelihood for the background $L(B)$ depends on which matting problem we are solving. For natural image matting, we use an analogous term to that of the foreground, setting w_i to $(1-\alpha_i)^2 g_i$ and substituting B in place of F in every term of equations (6), (7), and (8). For constant-color matting, we calculate the mean and covariance for the set of all pixels that are labelled as background. For difference matting, we have the background color at each pixel; we therefore use the known background color as the mean and a user-defined variance to model the noise of the background.

In this work, we assume that the log likelihood for the opacity $L(\alpha)$ is constant (and thus omitted from the maximization in equation (4)). A better definition of $L(\alpha)$ derived from statistics of real alpha mattes is left as future work.

Because of the multiplications of α with F and B in the log likelihood $L(C \mid F,B,\alpha)$, the function we are maximizing in (4) is not a quadratic equation in its unknowns. To solve the equation efficiently, we break the problem into two quadratic sub-problems. In the first sub-problem, we assume that α is a constant. Under this assumption, taking the partial derivatives of (4) with respect to F and B and setting them equal to 0 gives:

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & \Sigma_B^{-1} + I(1-\alpha)^2/\sigma_C^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \Sigma_F^{-1}\overline{F} + C\alpha/\sigma_C^2 \\ \Sigma_B^{-1}\overline{B} + C(1-\alpha)/\sigma_C^2 \end{bmatrix}, \qquad (9)$$

where I is a 3×3 identity matrix. Therefore, for a constant α, we can find the best parameters F and B by solving the 6×6 linear equation (9).

In the second sub-problem, we assume that F and B are constant, yielding a quadratic equation in α. We arrive at the solution to this equation by projecting the observed color C onto the line segment FB in color space:

$$\alpha \;=\; \frac{(C-B)\cdot(F-B)}{\|F-B\|^2}\,. \qquad (10)$$

where the numerator contains a dot product between two color difference vectors. To optimize the overall equation (4) we

alternate between assuming that α is fixed to solve for F and B using (9), and assuming that F and B are fixed to solve for α using (10). To start the optimization, we initialize α with the mean α over the neighborhood of nearby pixels and then solve the constant-α equation (9).

When there is more than one foreground or background cluster, we perform the above optimization procedure for each pair of foreground and background clusters and choose the pair with the maximum likelihood. Note that this model, in contrast to a mixture of Gaussians model, assumes that the observed color corresponds to exactly one pair of foreground and background distributions. In some cases, this model is likely to be the correct model, but we can certainly conceive of cases where mixtures of Gaussians would be desirable, say, when two foreground clusters can be near one another spatially and thus can mix in color space. Ideally, we would like to support a true Bayesian mixture model. In practice, even with our simple exclusive decision model, we have obtained better results than the existing approaches.

4. Results and comparisons

We tried out our Bayesian approach on a variety of different input images, both for blue-screen and for natural image matting. Figure 2 shows four such examples. In the rest of this section, we discuss each of these examples and provide comparisons between the results of our algorithm and those of previous approaches. For more results and color images, please visit the URL listed under the title.

4.1. Blue-screen matting

We filmed our target object, a stuffed lion, in front of a computer monitor displaying a constant blue field. In order to obtain a ground-truth solution, we also took radiance-corrected, high dynamic range [3] pictures of the object in front of five additional constant-color backgrounds. The ground-truth solution was derived from these latter five pictures by solving the overdetermined linear system of compositing equations (1) using singular value decomposition.

Both Mishima's algorithm and our Bayesian approach require an estimate of the background color distribution as input. For blue-screen matting, a preliminary segmentation can be performed more-or-less automatically using the Vlahos equation (2) from Section 2. Setting a_1 to be a large number generally gives regions of pure background (where $\alpha \leq 0$), while setting a_1 to a small number gives regions of pure foreground (where $\alpha \geq 1$). The leftmost image in the middle row of Figure 2 shows the preliminary segmentation produced in this way, which was used as input for both Mishima's algorithm and our Bayesian approach.

In Figure 3, we compare our results with Mishima's algorithm and with the ground-truth solution. Mishima's algorithm exhibits obvious "blue spill" artifacts around the boundary, whereas our Bayesian approach gives results that appear to be much closer to the ground truth.

4.2. Natural image matting

Figure 4 provides an artificial example of "natural image matting," one for which we have a ground-truth solution. The input image was produced by taking the ground-truth solution for the previous blue-screen matting example, compositing it over a (known) checkerboard background, displaying the resulting image on a monitor, and then re-photographing the scene. We then attempted to use four different approaches for re-pulling the matte: a simple difference matting approach (which takes the difference of the image from the known background, thresholds it, and then blurs the result to soften it); Knockout; the Ruzon and Tomasi algorithm, and our Bayesian approach. The ground-truth result is repeated here for easier visual comparison. Note the checkerboard artifacts that are visible in Knockout's solution. The Bayesian approach gives mattes that are somewhat softer, and closer to the ground truth, than those of Ruzon and Tomasi.

Figure 5 repeats this comparison for two (real) natural images (for which no difference matting or ground-truth solution is possible). Note the missing strands of hair in the close-up for Knockout's results. The Ruzon and Tomasi result has a discontinuous hair strand on the left side of the image, as well as a color discontinuity near the center of the inset. In the lighthouse example, both Knockout and Ruzon-Tomasi suffer from background spill. For example, Ruzon-Tomasi allows the background to blend through the roof at the top center of the composite inset, while Knockout loses the railing around the lighthouse almost completely. The Bayesian results exhibit none of these artifacts.

5. Conclusions

In this paper, we have developed a Bayesian approach to solving several image matting problems: constant-color matting, difference matting, and natural image matting. Though sharing a similar probabilistic view with Ruzon and Tomasi's algorithm, our approach differs from theirs in a number of key aspects; namely, it uses (1) MAP estimation in a Bayesian framework to optimize α, F and B simultaneously, (2) oriented Gaussian covariances to better model the color distributions, (3) a sliding window to construct neighborhood color distributions that include previously computed values, and (4) a scanning order that marches inward from the known foreground and background regions. To sum up, our approach has an intuitive probabilistic motivation, is relatively easy to implement, and compares favorably with the state of the art in matte extraction.

In the future, we hope to explore a number of research directions. So far, we have omitted using priors on alpha. We hope to build these priors by studying the statistics of ground truth alpha mattes, possibly extending this analysis to evaluate spatial dependencies that might drive an MRF approach

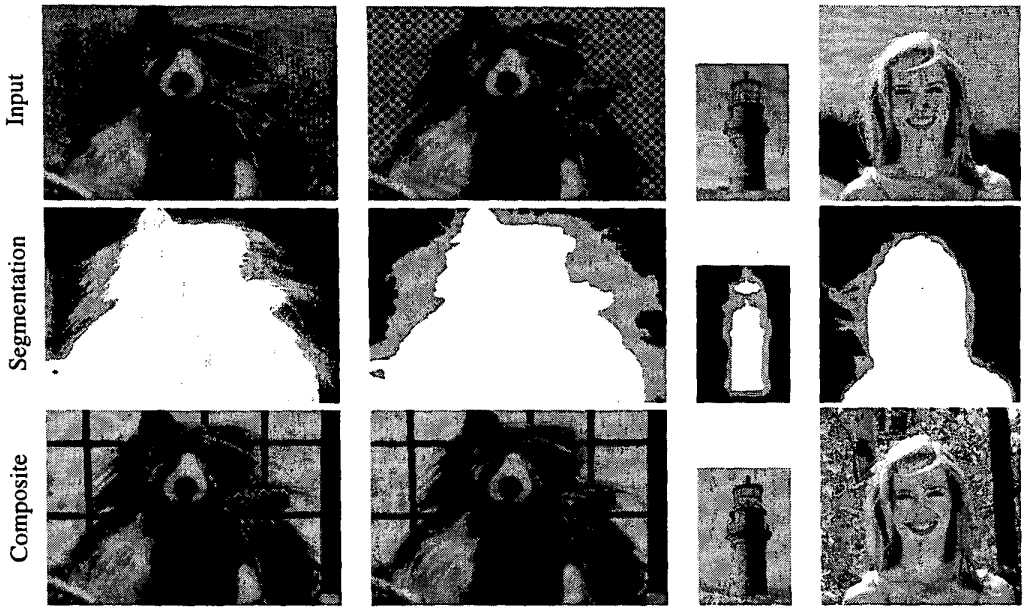

Figure 2 Summary of input images and results. Input images (top row): a blue-screen matting example of a toy lion, a synthetic "natural image" of the same lion (for which the exact solution is known), and two real natural images, (a lighthouse and a woman). Input segmentation (middle row): conservative foreground (white), conservative background (black), and "unknown" (grey). The leftmost segmentation was computed automatically (see text), while the rightmost three were specified by hand. Compositing results (bottom row): the results of compositing the foreground images and mattes extracted through our Bayesian matting algorithm over new background scenes. (Lighthouse image and the background images in composite courtesy Philip Greenspun, http://philip.greenspun.com. Woman image was obtained from Corel Knockout's tutorial, Copyright © 2001 Corel. All rights reserved.)

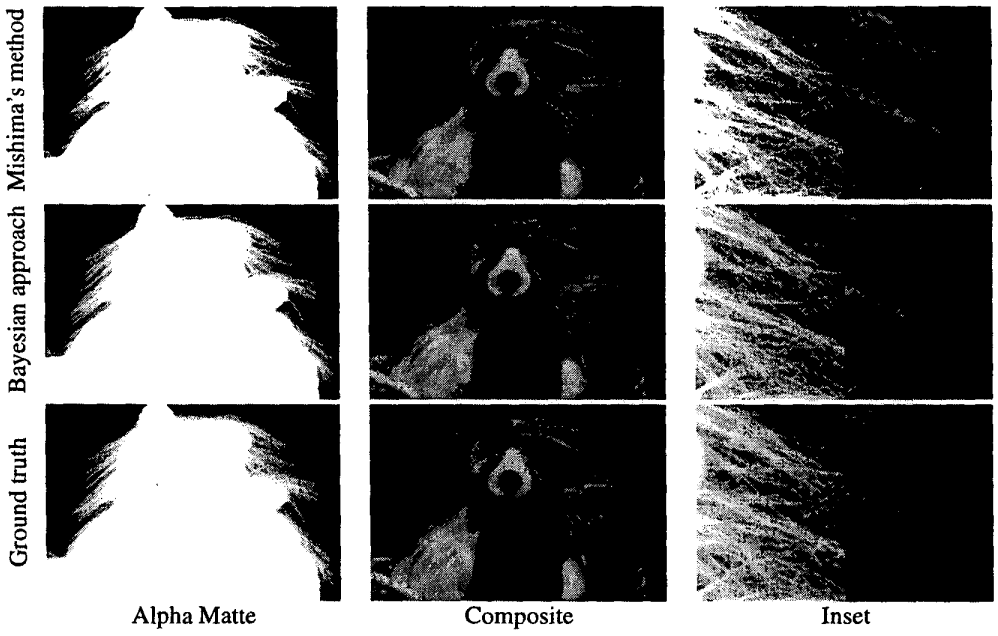

Figure 3 Blue-screen matting of lion (taken from leftmost column of Figure 2). Mishima's results in the top row suffer from "blue spill." The middle and bottom rows show the Bayesian result and ground truth, respectively.

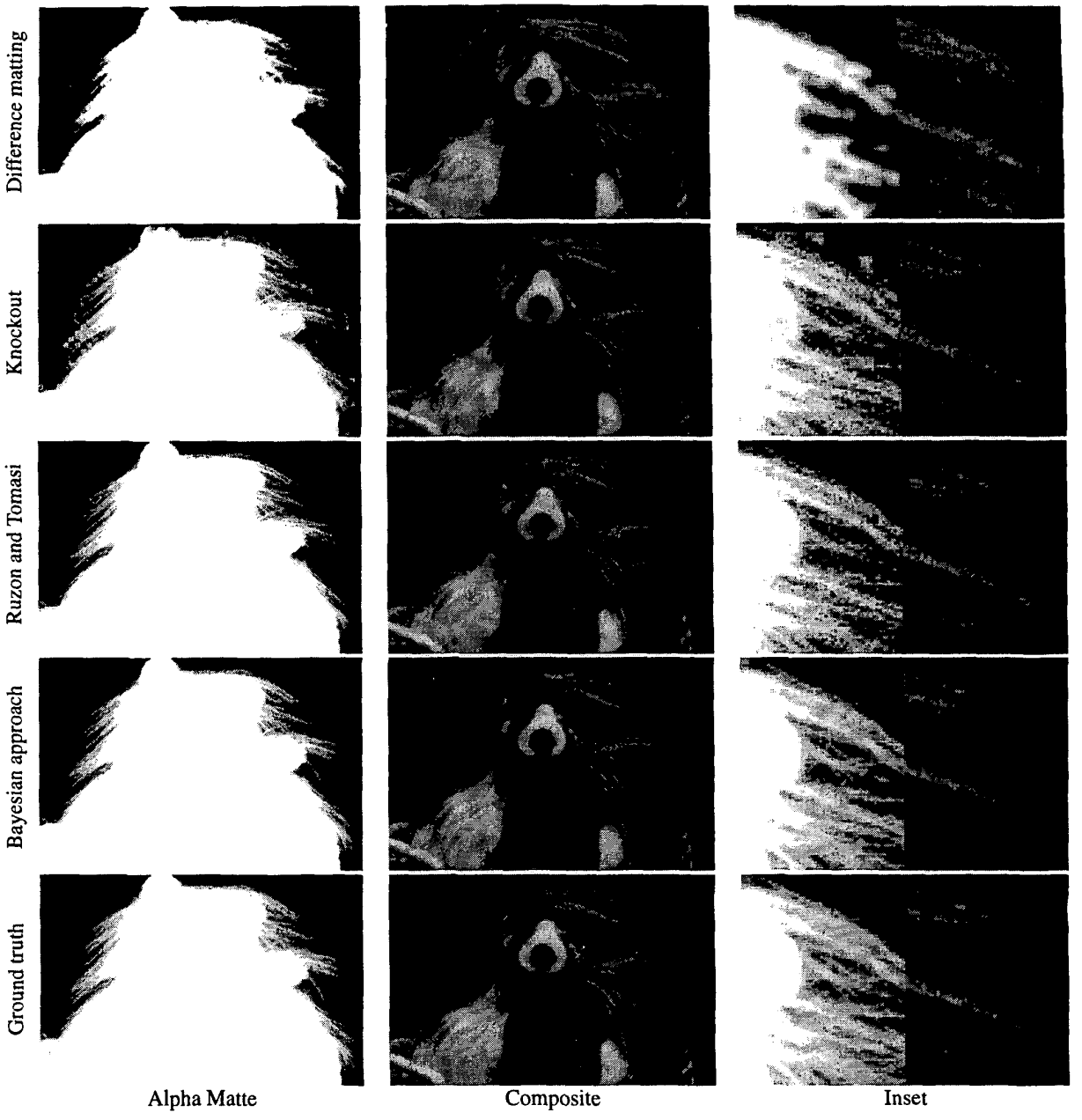

Figure 4 "Synthetic" natural image matting. The top row shows the results of difference image matting and blurring on the synthetic composite image of the lion against a checkerboard (column second from left in Figure 2). Clearly, difference matting does not cope well with fine strands. The second row shows the result of applying Knockout; in this case, the interpolation algorithm poorly estimates background colors that should be drawn from a bimodal distribution. The Ruzon-Tomasi result in the next row is clearly better, but exhibits a significant graininess not present in the Bayesian matting result on the next row or the ground-truth result on the bottom row.

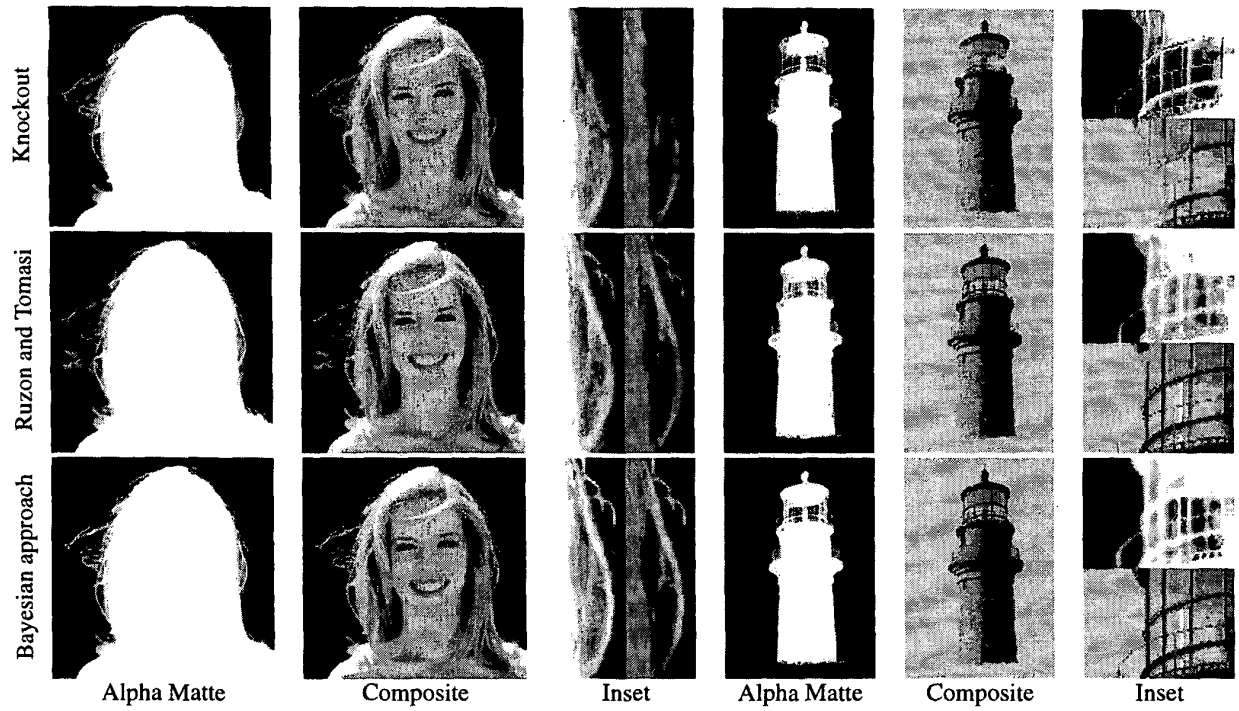

Figure 5 Natural image matting. These two sets of photographs correspond to the rightmost two columns of Figure 2, and the insets show both a close-up of the alpha matte and the composite image. For the woman's hair, Knockout loses strands in the inset, whereas Ruzon-Tomasi exhibits broken strands on the left and a diagonal color discontinuity on the right, which is enlarged in the inset. Both Knockout and Ruzon-Tomasi suffer from background spill as seen in the lighthouse inset, with Knockout practically losing the railing.

to image matting. Next, we hope to extend our framework to support mixtures of Gaussians in a principled way, rather than arbitrarily choosing among paired Gaussians as we do currently. Finally, we plan to extend our work to video matting with soft boundaries.

Acknowledgments

The authors would like to thank Ja-Chi Wu for his assistance with creating the figures for this paper. This work was supported by NSF grant CCR-987365 and by an Intel equipment donation.

References

[1] A. Berman, A. Dadourian, and P. Vlahos. Method for removing from an image the background surrounding a selected object. U.S. Patent 6,134,346, 2000.

[2] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6,134,345, 2000.

[3] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97*, pages 369–378, Aug. 1997.

[4] R. Fielding. *The Technique of Special Effects Cinematography*. Focal/Hastings House, London, 3rd edition, 1972.

[5] Y. Mishima. Soft edge chroma-key generation based upon hexoctahedral color space. U.S. Patent 5,355,174, 1993.

[6] T. Mitsunaga, T. Yokoyama, and T. Totsuka. Autokey: Human assisted key extraction. In *SIGGRAPH 95*, pages 265–272, August 1995.

[7] M. T. Orchard and C. A. Bouman. Color Quantization of Images. *IEEE Transactions on Signal Processing*, 39(12):2677–2690, December 1991.

[8] T. Porter and T. Duff. Compositing digital images. In *SIGGRAPH 1984*, pages 253–259, July 1984.

[9] R. J. Qian and M. I. Sezan. Video background replacement without a blue screen. In *ICIP 1999*, pages 143–146, October 1999.

[10] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *CVPR 2000*, pages 18–25, June 2000.

[11] A. R. Smith and J. F. Blinn. Blue screen matting. In *Proceedings of SIGGRAPH 96*, pages 259–268, Aug. 1996.

[12] B. A. Wallace. Automated production techniques in cartoon animation. Master's thesis, Cornell University, 1982.